# SINGLE-CHIP MICROCOMPUTER AND BOOT REGION SWITCHING METHOD THEREOF

## BACKGROUND OF THE INVENTION

### 1. Field of the Invention

The present invention relates to a single-chip microcomputer having a CPU and a nonvolatile memory and, more particularly, to a boot region switching method of, when a certain region of a nonvolatile memory is designated as a boot region where a boot program is stored, switching other region to be a new boot region.

### 2. Description of the Related Art

Single-chip microcomputer (or single-chip micro) is a semiconductor device in which a storage region where a control program is stored and a CPU (central processing unit) for executing operation based on the control program are formed on one semiconductor chip.

In order to facilitate modification of a control program, some of such single-chip microcomputers employ nonvolatile memories including a flash memory as a memory for storing the control program.

Single-chip microcomputer mounted with such a flash memory, even when a control program stored in the flash memory needs to be modified, enables the once stored control program to be rewritten.

Among methods of rewriting such a control program are a writer rewriting mode for rewriting a control

program by using a flash writer and a self-programming
mode for rewriting a nonvolatile memory at the stage of
user's use. The writer rewriting mode is a mode of
controlling rewriting of a flash memory by starting

5    instruction execution from a boot memory to communicate
with the outside, while a user mode is a mode of
executing a user region by similarly booting from a boot
memory and then selecting a start address. In the self-
programming mode, a boot program and other instructions

10   necessary for rewriting are transferred from a
nonvolatile memory or from the outside to a RAM to
rewrite the nonvolatile memory based on the instruction
read from the RAM. In the self-programming mode, only
when rewriting a region containing a boot block, a boot

15   program is required as writing data. Boot program here
represents a program for booting, which is a program to
be executed first at the time of system booting after
the system is reset.

At the time of rewriting a certain memory block

20   of a nonvolatile memory by the self-programming mode as
described above or at the time of replacing a boot
program with a new boot program, boot region switching
processing is executed of rewriting a boot program
stored in the memory block in question into another

25   memory block or storing a new boot program in another
memory block to switch the region in which the boot
program is stored to a new boot region.

However, simply storing a boot program stored in a user region of a certain memory block into a RAM, erasing an original boot program and storing the boot program read from the RAM in a user region of another
5     memory block might result in that a single-chip microcomputer system can not continue with normal operation due to instantaneous cut-off of power or the like occurring during erasure of a user region of a nonvolatile memory in which programs including a boot
10    program (hereinafter referred to as a boot program) are stored. In another case, instantaneous cut-off of power might occur during new program writing after erasure to result in incomplete writing. In such cases, even the system is reset to boot again, there are cases where a
15 ·   boot program has not been normally rewritten to disable re-booting. Therefore, the self-programming mode of rewriting a nonvolatile memory at a stage of user's use has a problem that a user region in which a boot program is stored can not be erased safely.

20    As a countermeasure which eliminates the problem, techniques for safely rewriting a user region in which a boot program is stored are disclosed in Japanese Patent Laying-Open (kokai) No. Heisei 8-255084 and Japanese Patent Laying-Open (kokai) No. Heisei 10-149282.
25    Employed here at the time of rewriting a program in a boot area is copying an old boot program into a free user region to save the same, erasing storage contents

of the boot area in question and then writing a new boot program into the area, and after confirming the end of the writing, erasing the old boot program copied into the free user region. Therefore, even when there occurs

5    a situation where writing of a new program can not be executed completely due to instantaneous cut-off of power or the like, the system can be booted again by the old boot program to prevent the system from entering an unrecoverable state.

10   With the above-described methods, however, the number of actions is large such as copying of an old boot program into a free area and erasure of an old boot program when it becomes unnecessary, resulting in making procedures complicated and time-consuming.

15   Conventional single-chip microcomputer which solves the above-described problems is recited in Japanese Patent Laying-Open (kokai) No. 2001-195241.

The conventional single-chip microcomputer is designed to have a boot area designation flag provided

20   for each user region to enable a boot program stored in a nonvolatile memory to be safely rewritten in an on-board mode at a stage of user's use by a less number of actions. Structure of the conventional single-chip microcomputer is shown in Fig. 4.

25   Boot area designation flag, which is provided for each memory block, is a flag for indicating that a boot program is stored in a user region of the memory block

in question.

The conventional single-chip microcomputer includes, as illustrated in Fig. 4, a nonvolatile memory 41, a CPU 2, a boot memory 3, a region switching flag 4 and a RAM 5.

The nonvolatile memory 41, in which programs including a boot program are stored, is divided into two memory blocks 161 and 162. The memory block 161 is formed of a user region A for writing data such as a program code by a user and a region for storing a boot area designation flag A corresponding thereto and the memory block 162 is formed of a user region B and a region for storing a boot area designation flag B corresponding thereto.

The CPU 2 is a main processor which conducts control and operation according to a program stored in the nonvolatile memory 41. The CPU 2 conducts setting of the region switching flag 4 based on the contents of a boot area designation flag. The boot memory 3 has a function of booting the system according to a designated control mode. After the system is booted up, among the programs stored in the nonvolatile memory 41, the boot program is first processed. Rewriting of the boot program is executed by a flash writer when the computer is handled singly by a factory and is executed in a self mode when in an on-board mode at the stage of user's use. The region switching flag 4 has a function of

designating a user region in which the boot program is stored at the time of system booting. The RAM 5 temporality stores necessary data in response to an instruction from the CPU 2.

With reference to the flow chart of Fig. 5, description will be made of a procedure of rewriting programs including a boot program in a user region in a self-programming mode in thus structured conventional single-chip microcomputer when the user region A of the memory block 161 is designated for booting (boot area designation flag A = "0", the boot area designation flag B = "1"). The nonvolatile memory 41 is designed that, when data contents are erased, "1" is stored in all the erased bits.

Here, represent the user region and the boot area designation flag of the memory block 161 as the user region A and the boot area designation flag A, respectively, and the user region and the boot area designation flag of the memory block 162 as the user region B and the boot area designation flag B, respectively.

First, erase the storage contents of the user region B and the boot area designation flag B into which a boot program is newly written (Step 201). Then, write the new boot program into the user region B (Step 202). Next, write data so as to make the boot area designation flag B have boot designation (data "0") (Step 203).

Then, erase the storage contents of the user region A and the boot area designation flag A (Step 204). At this time, no write to the boot area designation flag A is conducted to make the data "1" remain the same. Next, write other program than the boot program into the erased user region A according to the necessity.

In the user mode, when the system is booted by resetting, the program is executed from the boot memory 3, based on which program values of the boot area designation flag A and the boot area designation flag B are read to determine which of the user region A and the user region B is an area including the new boot program and write data of the relevant user region into the region switching flag 4. Set the determination result at the region switching flag 4 to read storage contents branched into the user regions in which the new boot program is stored at the time of system booting.

When instantaneous cut-off of power occurs after the end of Step 203 and before Step 204 ends, a plurality of user regions might be designated as a boot region. In such a case, any of the user regions, the user region A, for example, should be preferentially designated as a boot region.

In the conventional single-chip microcomputer, since setting a boot area designation flag and a region switching flag enables a user region including a boot program for booting the system to be switched, thereby

enabling boot program rewriting in a nonvolatile memory to be executed always with the old boot program existing, even when rewriting fails to be completed due to instantaneous cut-off of power or the like at the time of rewriting in the self-mode, recovery is possible by further rewriting and rewriting is enabled with a less number of actions in a short period of time.

In such a conventional single-chip microcomputer, however, when instantaneous power cut-off occurs during erasure of the user region B, that is, during the processing of Step 201 in Fig. 5, the value of the boot area designation flag B becomes indefinite to have a valid value depending on situations. At this stage, however, processing of Step 202 for writing the new boot program into the user region B is yet to be completed. Therefore, when the system is reset to boot up, both the boot area designation flags A and B go valid and if setting is made here to preferentially designate the user region B as a boot region, such a problem occurs that a user region in which no boot program is stored might be designated as a boot region.

In a flash memory, in particular, pre-write is conducted of, before erasing written data, once executing writing to make a voltage level held in all the memory cells be the same level. Therefore, when instantaneous power cut-off or the like occurs during pre-write of a certain memory block, a boot area

designation flag of the memory block might attain "1" which should be originally "0", or attain "0" which should be originally "1".

As a result, in the conventional single-chip microcomputer shown in Fig. 4 as well, such a problem possibly occurs that a memory block in which a boot program is stored can not be specified depending on timing when instantaneous cut-off of power occurs, which disables booting from the boot program. Then, when booting from the boot program is not executed, system operation might not be conducted normally to have erroneous operation.

The above-described conventional single-chip microcomputers have shortcomings that instantaneous power cut-off or the like occurring during boot program rewriting processing in the self-programming mode makes booting from the boot program impossible to have erroneous operation.

## SUMMARY OF THE INVENTION

An object of the present invention is to provide a single-chip microcomputer capable of booting from a boot program without fail even when instantaneous power cut-off or the like occurs during boot region switching processing in self-programming.

According to the first aspect of the invention, a single-chip microcomputer comprises a non-volatile

memory having a plurality of memory block pairs as a
combination of a plurality of memory blocks each formed
of a user region for storing a program code by a user
and a region for storing a boot area designation flag
5    indicating that a boot program to be executed first at
the time of system booting is stored in the user region,
in which the boot program is stored in a user region of
a memory block of a specific pair of memory blocks at an
initial state where no boot region switching processing
10   is conducted, a region switching flag for indicating in
which memory block pair the boot program is stored in
the user region among the plurality of pairs of memory
blocks, and a control element for, when designating
other pair of memory blocks not designated as a boot
15   region as a new boot region, storing a value obtained by
subtracting a predetermined value from an initial value
set at the boot area designation flag of the memory
block of all the memory block pairs in the boot area
designation flag of each memory block of the other
20   memory block pair and at the time of booting the system,
if the values of the boot area designation flags in the
memory blocks of the other memory block pair are equal
and not the initial value, determining that the boot
program is stored in the other memory block pair to set
25   the region switching flag.

In the preferred construction, the control
element, when the values of the boot area designation

flags in the memory blocks of the other memory block pair are different or the initial value, determines that the boot area designation flag in question is invalid to determine that the boot program is stored in the user region of the memory block of the specific memory block pair.

In another preferred construction, the control element, when the values of the boot area designation flags in the memory blocks of the other memory block pair are equal and not the initial value, if values of the boot area designation flags in the memory blocks of the specific memory block pair are different, determines that the boot area designation flag in the memory block of the specific memory block pair is invalid to determine that the boot program is stored in the other memory block pair.

In another preferred construction, the control element, when the values of the boot area designation flags in the memory blocks of the other memory block pair are equal and not the initial value, if values of the boot area designation flags in the memory blocks of the specific memory block pair are equal, determines that the boot program is stored in a memory block of a memory block pair which stores a boot area designation flag whose value is smaller.

In another preferred construction, the control element, when the values of the boot area designation

flags in the memory blocks of the other memory block
pair are equal and not the initial value or "0",
determines that the boot program is stored in the other
memory block pair.

5 In another preferred construction, at the time of
erasure, each memory block of the memory block pairs of
the non-volatile memory has the user region and the boot
area designation flag erased simultaneously.

According to the second aspect of the invention,
10 a single-chip microcomputer comprises a non-volatile
memory having a plurality of memory block pairs as a
combination of a plurality of memory blocks each formed
of a user region for storing a program code by a user
and a region for storing a boot area designation flag
15 indicating that a boot program to be executed first at
the time of system booting is stored in the user region,
in which the boot program is stored in a user region of
a memory block of a specific memory block pair at an
initial state where no boot region switching processing
20 is conducted, a region switching flag for indicating in
which memory block pair the boot program is stored in
the user region among the plurality of pairs of memory
blocks, and a control element for setting the region
switching flag, wherein the control element when
25 designating other memory block pair not designated as a
boot region as a new boot region, stores a value
obtained by subtracting a predetermined value from an

initial value set at the boot area designation flag of the memory block of all the memory block pairs in the boot area designation flag of each memory block of the other memory block pair and at the time of booting the

5    system, if the values of the boot area designation flags in the memory blocks of the other memory block pair are equal and not the initial value, determines that the boot program is stored in the other memory block pair,

when the values of the boot area designation

10   flags in the memory blocks of the other memory block pair are different or the initial value, determines that the boot area designation flag in question is invalid to determine that the boot program is stored in the user region of the memory block of the specific memory block

15   pair, when the values of the boot area designation flags in the memory blocks of the other memory block pair are equal and not the initial value, if values of the boot area designation flags in the memory blocks of the specific memory block pair are different, determines

20   that the boot area designation flag in the memory block of the specific memory block pair is invalid to determine that the boot program is stored in the other memory block pair, and when the values of the boot area designation flags in the memory blocks of the other

25   memory block pair are equal and not the initial value, if the values of the boot area designation flags in the memory blocks of the specific memory block pair are

equal, determines that the boot program is stored in a memory block of a memory block pair which stores a boot area designation flag whose value is smaller.

According to another aspect of the invention, in a single-chip microcomputer including a non-volatile memory having a plurality of memory block pairs as a combination of a plurality of memory blocks each formed of a user region for storing a program code by a user and a region for storing a boot area designation flag indicating that a boot program to be executed first at the time of system booting is stored in the user region, in which the boot program is stored in a user region of a memory block of a specific pair of memory blocks at an initial state where no boot region switching processing is conducted, and a CPU, a boot region designating method of the single-chip microcomputer of, when a user region of a memory block of a specific memory block pair is designated as a boot region in which the boot program is stored, switching a region of other memory block pair as a new boot region, comprising the steps of:

when designating other memory block pair not designated as a boot region as a new boot region, storing a value obtained by subtracting a predetermined value from an initial value set at the boot area designation flag of the memory block of all the memory block pairs in the boot area designation flag of each memory block of the other memory block pair, and

at the time of booting the system, if the values of the boot area designation flags in the memory blocks of the other memory block pair are equal and not the initial value, determining that the boot program is stored in the other memory block pair.

In the preferred construction, the boot region designating method of the single-chip microcomputer comprises the step of, when the values of the boot area designation flags in the memory blocks of the other memory block pair are different or the initial value, determining that the boot area designation flag in question is invalid to determine that the boot program is stored in the user region of the memory block of the specific memory block pair.

In another preferred construction, the boot region designating method of the single-chip microcomputer comprises the step of, when the values of the boot area designation flags in the memory blocks of the other memory block pair are equal and not the initial value, if values of the boot area designation flags in the memory blocks of the specific memory block pair are different, determining that the boot area designation flag in the memory block of the specific memory block pair is invalid to determine that the boot program is stored in the other memory block pair.

In another preferred construction, the boot region designating method of the single-chip

microcomputer comprises the step of, when the values of
the boot area designation flags in the memory blocks of
the other memory block pair are equal and not the
initial value, if values of the boot area designation

5      flags in the memory blocks of the specific memory block
pair are equal, determining that the boot program is
stored in a memory block of a memory block pair which
stores a boot area designation flag whose value is
smaller.

10          In another preferred construction, the boot
region designating method of the single-chip
microcomputer comprises the step of, when the values of
the boot area designation flags in the memory blocks of
the other memory block pair are equal and not the

15     initial value or "0", determining that the boot program
is stored in the other memory block pair.

According to another aspect of the invention, in
a single-chip microcomputer including a non-volatile
memory having a plurality of memory block pairs as a

20     combination of a plurality of memory blocks each formed
of a user region for storing a program code by a user
and a region for storing a boot area designation flag
indicating that a boot program to be executed first at
the time of system booting is stored in the user region,

25     in which the boot program is stored in a user region of
a memory block of a specific pair of memory blocks at an
initial state where no boot region switching processing

is conducted, and a CPU, a boot region designating method of the single-chip microcomputer of, when a user region of a memory block of a specific memory block pair is designated as a boot region in which the boot program

5      is stored, switching a region of other memory block pair as a new boot region, comprising the steps of when designating other memory block pair not designated as a boot region as a new boot region, storing a value obtained by subtracting a predetermined value from an

10     initial value set at the boot area designation flag of the memory block of all the memory block pairs in the boot area designation flag of each memory block of the other memory block pair, at the time of booting the system, if the values of the boot area designation flags

15     in the memory blocks of the other memory block pair are equal and not the initial value, determining that the boot program is stored in the other memory block pair, when the values of the boot area designation flags in the memory blocks of the other memory block pair are

20     different or the initial value, determining that the boot area designation flag in question is invalid to determine that the boot program is stored in the user region of the memory block of the specific memory block pair, when the values of the boot area designation flags

25     in the memory blocks of the other memory block pair are equal and not the initial value, if values of the boot area designation flags in the memory blocks of the

specific memory block pair are different, determining that the boot area designation flag in the memory block of the specific memory block pair is invalid to determine that the boot program is stored in the other

5     memory block pair, and when the values of the boot area designation flags in the memory blocks of the other memory block pair are equal and not the initial value, if values of the boot area designation flags in the memory blocks of the specific memory block pair are

10     equal, determining that the boot program is stored in a memory block of a memory block pair which stores a boot area designation flag whose value is smaller.

       According to another aspect of the invention, a boot region switching method of a microcomputer

15     including a non-volatile memory having a first memory block and a second memory block formed of a program region and a boot area designation flag region, of the microcomputer, comprising the steps of

       when storing a new boot program in the non-

20     volatile memory instead of the boot program, erasing data of the second memory block, storing the new boot program in the program region of the second memory block, storing data which are different from data stored in the boot area designation flag region of the first memory

25     block in the boot area designation flag region of the second memory block, and erasing data of the first memory block.

In the preferred construction, when reading the boot program, reading the boot area designation flag from the first memory block and the second memory block, executing the boot program stored in the first memory block or the new boot program stored in the second memory block based on a comparison result of the boot area designation flag.

In another preferred construction, the boot area designation flag region of the second memory block including a first boot area designation flag region and a second boot area designation flag region, instead of the step of storing data which are different from data stored in the boot area designation flag region of the first memory block in the boot area designation flag region of the second memory block, storing data which are different from data stored in the boot area designation flag region of the first memory block in the first boot area designation flag region of the second memory block, storing same data as data of the boot area designation flag region of the first memory block in the second boot area designation flag region of the second memory block.

According to the present invention, since boot area designation flags of the same contents are stored in a plurality of regions which can not be erased simultaneously, even when instantaneous power cut-off occurs to make one boot area designation flag have an

indefinite value, a value of other boot area designation flag is maintained to be normal. Therefore, detecting whether the two boot area designation flags are coincident with each other or not leads to determination whether the boot area designation flags are valid or invalid, so that erroneous operation can be prevented of designating a memory block in which no boot program is actually stored as a boot region.

Other objects, features and advantages of the present invention will become clear from the detailed description given herebelow.

## BRIEF DESCRIPITION OF THE DRAWINGS

The present invention will be understood more fully from the detailed description given herebelow and from the accompanying drawings of the preferred embodiment of the invention, which, however, should not be taken to be limitative to the invention, but are for explanation and understanding only.

In the drawings:

Fig. 1 is a block diagram showing a structure of a single-chip microcomputer according to a first embodiment of the present invention;

Fig. 2A is a diagram for use in explaining boot region switching processing in the single-chip microcomputer according to the first embodiment of the present invention;

Fig. 2B is a diagram for use in explaining boot region switching processing in the single-chip microcomputer according to the first embodiment of the present invention;

5      Fig. 2C is a diagram for use in explaining boot region switching processing in the single-chip microcomputer according to the first embodiment of the present invention;

Fig. 2D is a diagram for use in explaining boot 10     region switching processing in the single-chip microcomputer according to the first embodiment of the present invention;

Fig. 2E is a diagram for use in explaining boot region switching processing in the single-chip 15     microcomputer according to the first embodiment of the present invention;

Fig. 2F is a diagram for use in explaining boot region switching processing in the single-chip microcomputer according to the first embodiment of the 20     present invention;

Fig. 3 is a flow chart showing operation of the single-chip microcomputer illustrated in Fig. 1;

Fig. 4 is a block diagram showing a structure of a conventional single-chip microcomputer; and

25     Fig. 5 is a flow chart showing operation of the conventional single-chip microcomputer illustrated in Fig. 4.

## DESCRIPTION OF THE PREFERRED EMBODIMENT

The preferred embodiment of the present invention will be discussed hereinafter in detail with reference

5    to the accompanying drawings. In the following description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be obvious, however, to those skilled in the art that the present invention may be practiced

10   without these specific details. In other instance, well-known structures are not shown in detail in order to unnecessary obscure the present invention.

Next, embodiment of the present invention will be detailed with reference to the drawings. Fig. 1 is a

15   block diagram showing a structure of a single-chip microcomputer according to a first embodiment of the present invention. In Fig. 1, the same components as those in Fig. 4 are given the same reference numerals to omit their description.

20   The single-chip microcomputer according to the present embodiment differs from the conventional single-chip microcomputer shown in Fig. 4 in that the nonvolatile memory 41 is replaced by a nonvolatile memory 1. In the present embodiment, description will be

25   made of a case where a flash memory is used as a nonvolatile memory.

The nonvolatile memory 1 in the present

embodiment is divided into four memory blocks 61 to 64.
Then, each of the four memory blocks 61 to 64 is divided
into a user region for writing data such as a program
code by a user and a region for storing a boot area

5    designation flag.  In the present embodiment, description
will be made assuming that the boot area designation
flag is formed of 32 bits as an example.

In addition, assuming that boot designation which
indicates a memory block in which a boot program is

10   stored is conducted for every two memory blocks, boot
designation is made here by designating a pair of memory
blocks, the two memory blocks 61 and 62 or the memory
blocks 63 and 64.

While in a conventional single-chip microcomputer,

15   the boot area designation flag only indicates whether a
boot program is stored in a block memory in question or
not, in the single-chip microcomputer of the present
embodiment, based on a value of the boot area
designation flag, determination is made in which of the

20   memory block pair the boot program is stored, the memory
blocks 61 and 62 or the memory blocks 63 and 64.

The flash memory is designed such that when a
certain memory block is erased, a boot area designation
flag and data stored in a user region in the memory

25   block are erased simultaneously to make it impossible to
have only the data stored in any one of the regions
remain.  In addition, in any erasing method, erasure is

internally conducted for every one memory block without fail.

A CPU 2 in the present embodiment determines in which of the memory block pairs, the two memory blocks 61 and 62 or the memory blocks 63 and 64, a boot program is stored which is to be executed when the system is reset next based on the values of the boot area designation flags stored in the respective memory blocks 61-64 and sets the determination result at the flag region switching flag 4.

Next, processing of switching a boot region which is conducted in the single-chip microcomputer according to the present embodiment will be described with reference to Figs. 2A, 2B, 2C, 2D, 2E and 2F. In Figs. 2A to 2F, the value of the boot area designation flag "FFFFFFFF (H: hexadecimal notation)" is represented simply as "FF", "FFFFFFFE(H)" simply as "FE" and "00000000(H)" simply as "00". Then, the boot region switching processing is conducted by the CPU 2.

Here, description will be made with respect to a case where a boot program stored in the user region of the memory block 61 is rewritten into the user region of the memory block 63. An initial value of the boot area designation flag (value at the erased state) is "FFFFFFFF(H)".

State of the nonvolatile memory 1 before the boot region switching processing is conducted is shown in Fig.

2A. In Fig. 2A, the boot program is stored in the user region of the memory block 61 and each boot area designation flag of the memory blocks 61 to 64 indicates "FFFFFFFF(H)".

5      The boot region switching processing is conducted from the state shown in Fig. 2A by the following manner.

(1) First, erase the memory blocks 63 and 64 (Fig. 2B) to write a new boot program in the user region of the memory block 63 (Fig. 2C). The value of the boot

10     area designation flag obtained when erasure is conducted changes once from "FFFFFFFF" to "00000000" and then again to "FFFFFFFF".

(2) Next, as the boot area designation flag of the memory block 63, write "FFFFFFFE" which is a value

15     obtained by decrementing "FFFFFFFF" by 1 (Fig. 2D) and thereafter as the boot area designation flag of the memory block 64, similarly write "FFFFFFFE" which is a value obtained by decrementing "FFFFFFFF" by 1 (Fig. 2E).

(3) Then, lastly, erase the memory blocks 61 and

20     62 to rewrite the data by the same manner as conventional one. In this processing, no rewriting is made of the boot area designation flags of the memory blocks 61 and 62.

Next, with reference to the flow chart shown in

25     Fig. 3, description will be made of boot area determination processing in which the CPU 2 determines a pair of memory blocks where a boot program is stored

based on a boot area designation flag value, which processing is conducted after thus executing the boot region switching processing. Here, the boot area designation flags of the memory blocks 61 to 64 are

5     represented as boot area designation flags A to D, respectively.

First, after the boot region switching processing is completed, the CPU 2 reads the values of the boot area designation flags A to D from the memory blocks 61

10     to 64 (Step 101). Then, determination is made whether the value of the boot area designation flag C and that of the boot area designation flag D are equal (Step 102). When at Step 102 the values of the boot area designation flag C and the boot area designation flag D are not the

15     same, which means that processing of switching the boot region to the memory blocks 63 and 64 fails to end normally, determination is made that a normal boot program is stored in the user regions of the memory blocks 61 and 62 to conduct boot designation of the

20     memory blocks 61 and 62 (Step 107).

When at Step 102 the determination is made that the values of the boot area designation flag C and the boot area designation flag D are the same, determination is next made whether the values of the boot area

25     designation flags C and D are "00000000" or "FFFFFFFF" (Step 103). When at Step 103 the values of the boot area designation flags C and D are "00000000" or "FFFFFFFF",

determination is made that one abnormality or another occurs in the memory blocks 63 and 64 to conduct boot designation of the memory blocks 61 and 62 (Step 107).

Here, the reason why the determination can be made that one abnormality or another occurs when the values of the boot area designation flags C and D are "00000000" or "FFFFFFFF" is that when instantaneous power cut-off occurs during erasure processing of a certain memory block, it is highly probable that data being erased will be "FFFFFFFF" and when instantaneous power cut-off occurs during pre-writing in the course of erasure processing, it is highly probable that data being erased will be "00000000".

Then, when normal processing is being executed, it is hardly possible that the values of the boot area designation flags C and D attain "00000000" or "FFFFFFFF" because of the following reasons.

First, description will be made of a case where the values of the boot area designation flags C and D are "FFFFFFFF. In a case where the boot program is stored at an initial state in any of the memory blocks 61 and 62, the boot program being stored in the memory blocks 63 and 64 means that the boot region switching processing has been conducted at least once, whereby the values of the boot area designation flags C and D can not remain "FFFFFFFF". Conversely, the values of the boot area designation flags C and D being "FFFFFFFF"

means that the boot region switching processing has never been conducted, whereby determination can be made that the boot program is stored in the memory blocks 61 and 62.

Next, description will be made of a case where the values of the boot area designation flags C and D are "00000000". Although because the value of the boot area designation flag is decremented by one each from the initial value "FFFFFFFF" every time the boot region switching processing is conducted, it is possible to be "00000000" at the last, in a case, for example, where the boot area designation flag has 32 bits, "00000000" can not be attained unless the boot region switching processing is conducted $2^{32}$ times, so that it is impossible in practice to attain "00000000". As a result, when the value of the boot area designation flag is "00000000", the determination can be made that one abnormality or another occurs.

In other words, when all the bits forming the boot area designation flag are "0" or "1" except at the initial state, even if values of two boot area designation flags stored in a plurality of memory blocks coincide with each other, the CPU 2 determines that they are invalid.

When at Step 103, the values of the boot area designation flags C and D are neither "00000000" nor "FFFFFFFF", determination is made whether the value of

the boot area designation flag A and that of the boot area designation flag B coincide with each other (Step 104). When at Step 104 the value of the boot area designation flag A and that of the boot area designation

5 flag B fail to coincide with each other, because the processing of switching a boot region to the memory blocks 61 and 62 fail to end normally, the determination is made that a normal boot program is stored in the user regions of the memory blocks 61 and 62 to conduct boot

10 designation of the memory blocks 63 and 64 (Step 108).

When at Step 104 the determination is made that the value of the boot area designation flag A and that of the boot area designation flag B coincide with each other, determination is next made whether the values of

15 the boot area designation flag A and the boot area designation flag B are "00000000" or not (Step 105). When at Step 105 the values of the boot area designation flags A and B are "00000000", the determination is made that one abnormality or another occurs in the memory

20 blocks 61 and 62 for the same reason as described above to conduct boot designation of the memory blocks 63 and 64 (Step 108).

Then, when at Step 105 the values of the boot area designation flags A and B are not "00000000",

25 determination is made whether the values of the boot area designation flags A and B are larger than the values of the boot area designation flags C and D (Step

105). When at Step 105 the values of the boot area designation flags A and B are larger than the values of the boot area designation flags C and D, the determination is made that the boot program is stored in

5     the user regions of the memory blocks 63 and 64 to conduct boot designation of the memory blocks 63 and 64 (Step 108). Conversely, when at Step 105 the values of the boot area designation flags A and B are smaller than the values of the boot area designation flags C and D,

10    the determination is made that the boot program is stored in the user regions of the memory blocks 61 and 62 to conduct boot designation of the memory blocks 61 and 62 (Step 107).

The foregoing described processing enables the

15    single-chip microcomputer of the present embodiment to specify a memory block in which a boot program is stored to boot the system from the boot program without fail even when instantaneous power cut-off occurs at any timing during self-programming.

20    In the flow chart shown in Fig. 3, the comparison is made between the value of the boot area designation flag and "00000000" or "FFFFFFFF", which comparison processing reduces a possibility that the values of two boot area designation flags will become indefinite to

25    coincide with each other by chance due to instantaneous power cut-off or the like.

Description will be made with respect to, for

example, a case where at the initial state of the single-chip microcomputer shown in Fig. 1, the boot program is stored only in the user region of the memory block 61 separately from the user region of the memory block 62. In such a case, if the boot region switching processing has never been conducted, all the values of the boot area designation flags A to D are the initial value "FFFFFFFF". When the self-programming processing is conducted from this state to rewrite the contents of the memory block 62, if instantaneous power cut-off occurs before the rewriting of the memory block 62 ends, the value of the boot area designation flag B of the memory block 62 becomes an indefinite value.

In this case, designating the memory blocks 63 and 64 as a boot region for the reason that the values of the boot area designation flags C and D are coincidently "FFFFFFFF" invites erroneous operation because the boot program is stored in the user region of the memory block 61 in practice. For preventing such a situation, the values of the boot area designation flags C and D and "FFFFFFFF" are compared in order to designate the memory blocks 61 and 62 as a boot region even in such a case.

In the single-chip microcomputer of the present embodiment, because the boot area designation flags of the same contents are stored in a plurality of regions which can not be erased simultaneously, even if

instantaneous power cut-off occurs to make one of the
boot area designation flags have an indefinite value,
the value of the other boot area designation flag is
maintained at a normal value. Therefore, determination

5      whether two boot area designation flags are coincident
or not leads to determination whether the boot area
designation flag is valid or not, thereby preventing
erroneous operation of designating a memory block in
which no boot program is actually stored as a boot

10     region.

Consideration will be given, for example, where
during erasing processing of the memory blocks 63 and 64,
instantaneous power cut-off occurs to make the data in
the memory blocks 63 and 64 indefinite. In this case,

15     even if either one of the boot area designation flags of
the memory blocks 63 and 64 has a value smaller than
"FFFFFFFF", unless the value of the other boot area
designation flag has the same value, none of the memory
blocks 63 and 64 is erroneously designated as a boot

20     region. Probability that values of the two boot area
designation flags coincide with each other by chance is
$1/2^{32}$ in a case where the boot area designation flag has
32 bits, which is too low to actually occur.

In addition, also when instantaneous power cut-

25     off occurs during erasing processing of the memory
blocks 61 and 62 to make the data in the memory blocks
61 and 62 indefinite as shown in Fig. 2F, erroneous

operation can not occur in practice for the same reason.

Although the present embodiment has been described with respect to a case where the number of memory blocks is four for the simplicity of description, the present invention is not limited thereto and the number of the memory blocks should be at least not less than four to have the present invention be applied. In this case, it is necessary that boot area designation flags of the same contents are stored in a plurality of memory blocks.

Moreover, although in the present embodiment, a storage region is divided into four memory blocks, each of which memory blocks is further divided into a region for storing a boot area designation flag and a user region to store boot area designation flags of the same contents in two different regions, the same effect can be obtained by any arrangement of storing boot area designation flags of the same contents in not less than two regions which can not be erased simultaneously.

As described in the foregoing, according to the present invention, since boot area designation flags of the same contents are stored in a plurality of memory blocks which can not be erased simultaneously, even when instantaneous power cut-off occurs during the boot region switching processing in self-programming, a memory block in which a boot program is stored can be specified to boot the system from the boot program.

Although the invention has been illustrated and described with respect to exemplary embodiment thereof, it should be understood by those skilled in the art that the foregoing and various other changes, omissions and additions may be made therein and thereto, without departing from the spirit and scope of the present invention. Therefore, the present invention should not be understood as limited to the specific embodiment set out above but to include all possible embodiments which can be embodies within a scope encompassed and equivalents thereof with respect to the feature set out in the appended claims.